

It is important to keep in mind that testing your settings will not save them. The purpose of this button is to show how your experimentations are going, not to permanently commit those settings into the format. You will need to click the **Save** button to do so. Click **Cancel** if you have not made any changes you would like to keep.

### Do I need to update the projects that use this format?

In a word, no. Once the format is saved and you have returned to the compile overview screen, those changes you made will be used for any projects making use of this format when next they are compiled. Formats you create and store in “My Formats” are separate from projects entirely and they all use the same copy. However if you have added new Section Layouts, may need to revise your layout assignments, from the compile overview screen.

If you need to save your format to make an adjustment to your project’s compile settings, perhaps to tweak a Replacement or assign an item to a Layout you just created, remember to save your compile settings before returning to the Format Designer screen: hold down the **Opt** key and click the **Save** button.

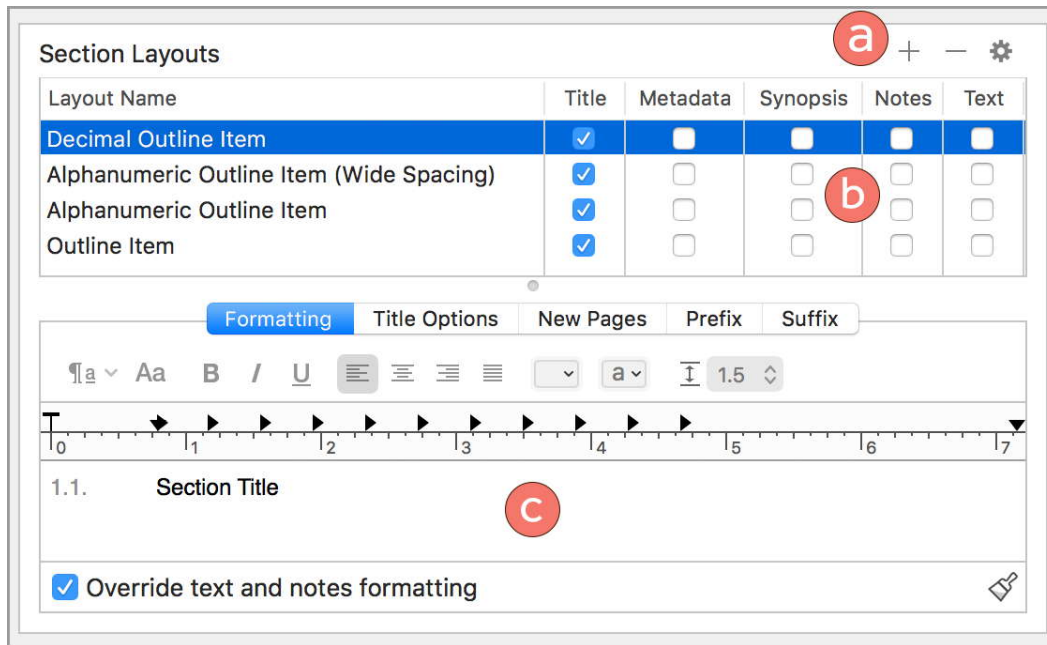
[Return to chapter](#) ↗

## 24.2 Section Layouts

The Section Types pane (available to all but the scriptwriting document types) is where you will define the available parts of a document, what text if any will be included within them and design how that text will be formatted (or structured, in the case of plain-text formats). In [Figure 24.2](#) we can see the list of layouts available from the “Enumerated Outline” compile format. It offers four choices for different numbering styles, listed in the upper half of the pane, and here we can see a preview of the “Decimal Outline Item” layout in the lower half.

The pane itself is split into two main sections one atop the other:

1. *Section Layout List*: here is where you will set which types of content should be used for each layout, such as title, text and other additional material. Clicking on an entry in this list will load its settings into the tabbed view in the lower half.
2. *Layout Editor*: everything that can be defined as part of a layout is set in this lower half of the pane. It is divided into several tabs which we will cover in the following sections.



**Figure 24.2:** The Section Layouts pane showing the “Enumerated Outline” layout list.

### 24.2.1 Section Layout List

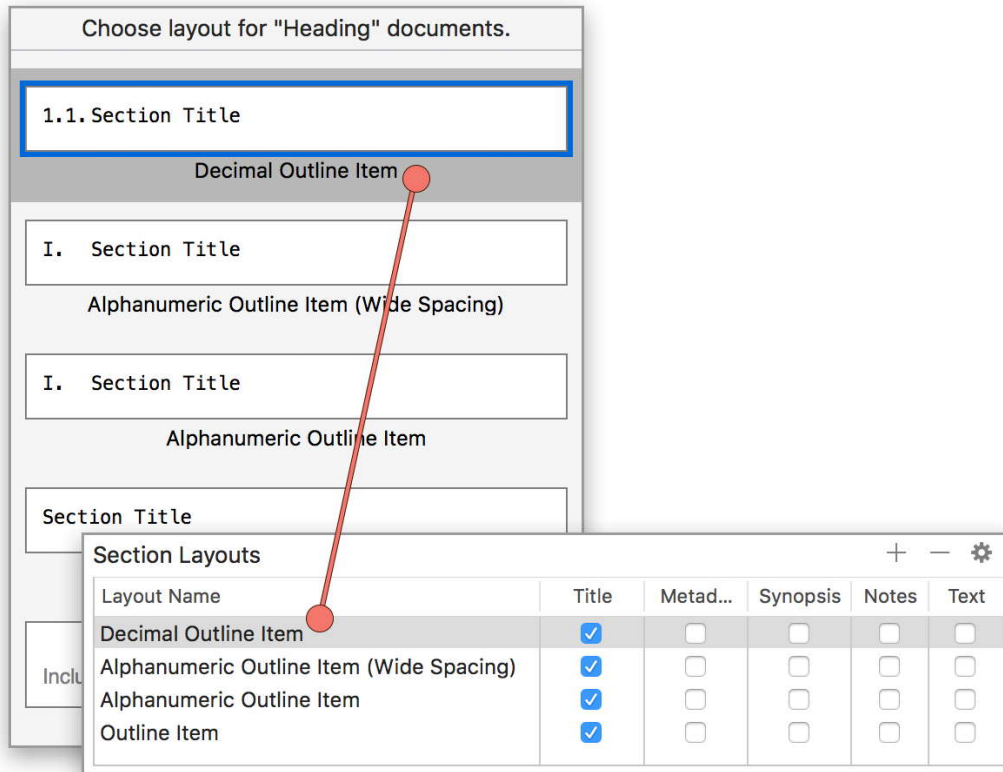
Each of the entries in this list are what will be provided in the Section Layouts column of the compile overview screen ([section 23.3](#)) as “tiles”. This can be seen most easily with a simple format, such as “Enumerated Outline” ([Figure 24.3](#)). The listed items in the pane (lower right excerpt) determines the names of the tiles and the order in which they appear in the assignment panel (behind and to the left in the figure).

Each row has a series of checkboxes on the right hand side, used to determine what content should be included with the layout. In these simple examples, where the goal is to print an indented outline of the heading structure of the draft, we are only including the **Title** for each layout. If we wanted to print a synopsis below each title, we would check off the **Synopsis** column.

### Adding New Layouts

To create a new layout:

1. Select the layout that most closely resembles the new one you’d like to create. You’ll be able to change everything about it, but if it is just a variation on a chapter heading it would be easiest to start from another chapter heading layout.
2. Click the **+** button in the upper right-hand corner.
3. Give the new layout a name and click elsewhere to confirm your change.



**Figure 24.3:** Section Layouts defines the layout tiles available in the assignment area of the compile overview screen.

## Renaming and Managing Layouts

Changing their order: the order of layouts isn't of great significance, but they will be listed in the order they appear within this list, when assigning types to layouts in the compile overview screen. If you want to change their order, just drag and drop.

To rename a layout: double-click on its name in the "Layout Name" column. This will not affect any projects using that layout for their types, they will automatically adjust to using the new name.

## Deleting a Layout


To remove a layout you no longer want in the format, select the layout from the listing and click the **–** button. This action cannot be undone, but you can of course close the format editor without saving changes by clicking the **Cancel** button along the bottom.

## Layout Content Columns


In all cases, these checkboxes merely add or remove the content itself from the layout. The formatting of that content, embellishment of it (such as adding deci-

mal numbering to the title) and so forth are done using the tabbed settings below this list.

**Title** Prints the binder title of each item assigned to use this layout. The title will typically be placed on a line of its own at the very top of the section, much like a heading or chapter break would be. This checkbox is only used to bring the binder title into the content—if you do not intend to do so, perhaps only printing the word “Chapter” followed by a number, then the checkbox is not necessary. An item can still have a *heading* without this checkbox.


Binder item titles will by default only include items that have been explicitly named—not adaptive titles being dynamically generated from synopses or text content ([section 7.3](#)). If you would rather have the adapted name used as a formal title, click the  button and enable the **Include placeholder titles for untitled items** option.

Within that same palette: items without explicit names will still go on to acquire any other heading the layout otherwise generates. If the layout prints “Chapter One” with the Title on a following line, the adaptively named item would simply print “Chapter One”. To have the entire heading omitted for items that are unnamed in the binder, enable the **Do not add prefix or suffix to placeholder titles** option.

When working with a Markdown-based format, by default titles will also automatically have hashes printed around them. You will see one hash pair appear as a preview, but the default behaviour is for the hash levels to be increased automatically depending upon outline depth. If you prefer titles only have a hashmark in front, click on the  button and disable the **Add closing hashes to titles** option.

**Metadata** Includes a stock block of text below the title containing all metadata associated with the item assigned to this layout. In the preview area, you will see a simple “Metadata: Listed Here” marker that can be used to format metadata lines in general. The number of lines will depend on each item—for example every item will have a created and modified date, but only say may have a line printing keywords. Each line of metadata will be preceded by a Tab, and can thus have its indent adjusted via changing the tab stop.


**Synopsis** The synopsis for the item will be printed and can be formatted as a paragraph in the preview area below.

**Notes** Any inspector Document Notes associated with the item will be printed before the main text. If you would prefer they fall after the content, click the  button in the upper right-hand corner and select the **Place notes after main text** option.

Notes can be optionally cleaned up to a consistent formatting along with (though independently) from the main text, by checking the **Override text and notes formatting** checkbox at the bottom of the pane.

**Text** The main text body of the item, such as the content of a section or chapter. Some layouts may not use this option if they intended only to be applied to types of items that serve as headings in the outline, or in some cases they may be omitted to achieve an effect, such as in the “Enumerated Outline” format, where the goal is to print an indented outline of topics rather than the entire work.

As with Notes, the main text can have its formatted cleaned and made uniform with the **Override text and notes formatting** checkbox.

If you include more than one type content other than **Title** and **Text**, sub-headings will be inserted automatically to announce the type of content. For example inserting “Synopsis” and “Notes” will add respective headings above the content. These headings can be formatted collectively to taste, but if you would rather they not appear, click the  button and disable the **Insert subtitles between text elements** option.

## Duplicating Settings Between Layouts

With so many options available in regards to the specific Layout itself it would be useful to have management tools for copying these settings from one layout to another. Say you’ve set up a meticulously designed layout for chapter breaks, and realise that with the exception of the title prefix, you want everything else to be applied to a variation use for interludes. You can use the following techniques for copying some or all settings (and remember that when creating new layouts you can always select the one you’d like to copy as a basis, before clicking the **+** button).

**Copy and paste portions of formatting** If all you want to do is copy the formatting you’ve applied from one element to another in a different area (even in a different row), click anywhere within the sample element you wish to copy from and then use the standard formatting copy and paste tools for doing so:

- **Format ▶ Paragraph ▶ Copy Paragraph Attributes** (**⌘C**) and **Paste Paragraph Attributes** (**⌘V**).
- **Format ▶ Font ▶ Copy Font** (**⌘C**) and **Paste Font** (**⌘V**).
- **Format ▶ Copy Formatting** (**⌘⇧C**) and **Paste Formatting** (**⌘⇧V**).

The following two tools are Layout row level actions; you must have a row in the structural table selected in order for them to work.


**Copy and Paste All Settings Between Rows** The standard commands for Copy and Paste can be used to bulk transfer all settings for both the structural table (whether title, notes, etc. export) and all settings below.

This action will also copy any settings that are assigned to this Layout from within the Separators pane ([section 24.4](#)).

**Setting the font for all elements** To set the base font for all elements within the formatting editor, select the row for the type and level you wish to edit and use **Format ▶ Font ▶ Show Fonts (⌘T)** to bring up the font palette. Any changes made here will impact every single element uniformly, so save this method for the very beginning of the customisation process. You will lose variant, size differences, and other characteristics that have already been applied to various elements.

This method will not work if elements of the formatting area cannot be modified, such as when the **Override text and notes formatting** checkbox is disabled.

## 24.2.2 Global Section Layout Options

In the upper right-hand corner, by the buttons used to add and remove Layouts, you will find a  button with a few options that will impact how all section layouts work.

**Include placeholder titles for untitled items** Those binder items that have been left untitled can optionally use the adaptive name generated from their content or synopses, as they are shown in the binder and outliner. The default behaviour is for these items to never show titles, even if they are assigned to a section layout that chiefly exists to generate a title.

**Do not add prefix or suffix to placeholder titles** When untitled documents are encountered, if the section layout they are assigned to generates a generic title (such as a chapter number), then that part of the title will still be used. Thus an untitled document may still have a structural presence in the final result. If you would prefer untitled elements be entirely anonymous and not contribute to the structure, then enable this option.

**Insert subtitles between text elements** If more than one type of content (excluding the **Title**) is included for a layout, then Scrivener will insert subtitles between those elements to help set them apart. These titles can be formatted like everything else in the formatting editor below. You only need to edit the formatting of one subtitle to impact them all. Disable this option if you would prefer to a seamless approach.

**Place notes after main text** The default is to place any Inspector notes above the main text for the item being compiled. When checked, notes will be placed below the main text area instead.

**Add closing hashes to titles** This option only pertains to the Markdown-based formats. When adding formal titles to the document, Scrivener will enclose the title in hashes:

```
### This is a Third Level Document ###
```

Disable this option to use the following format:

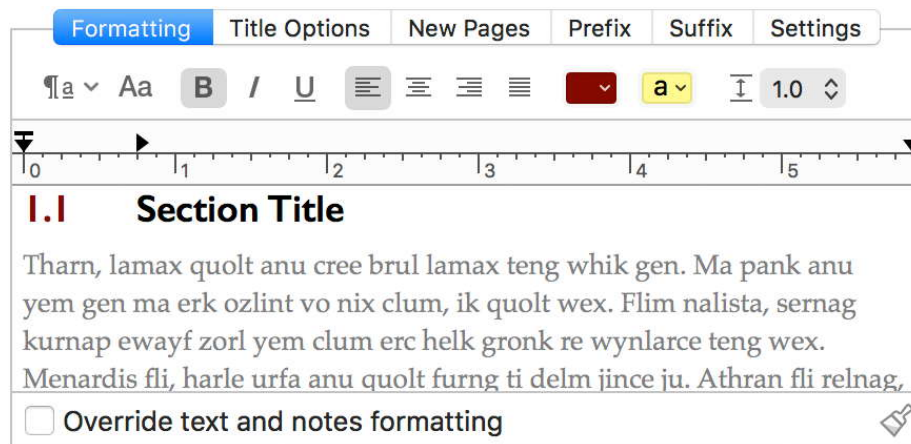
```
### This is a Third Level Document
```

### 24.2.3 Changing How a Layout Works

When you select a Layout in the list, its settings will be loaded into the tabbed configuration area below (showing a preview by default). You can thus switch between layouts and easily compare settings between them.

#### Formatting

The “Formatting” tab serves both as a preview of most of the settings that can be applied in the other tabs, and as a way to format the various pieces of content that can be included as part of a layout.



**Figure 24.4:** The Section Layouts “Formatting” tab is where you will set up the look of a layout.

The text itself cannot be directly edited, it merely serves as a template for what will be inserted by each binder item assigned to use this Layout. Instead, the preview comprises multiple *elements* that can be clicked within to have their formatting adjusted. In [Figure 24.4](#) we can see three such elements:

1. The title prefix, formatted in brown text, inserted from the “Title Options” tab.
2. The title itself, being inserted by the **Title** checkbox in the layout list above.



3. The main text body, which in this case cannot be formatted because the **Override text and notes formatting** checkbox below this preview area isn't ticked.

Formatting within the preview area is done in a very similar fashion to how would format text in the main editor, except you are working with *elements*, not literal text. You only need position the cursor anywhere within the “Section Title” element to fully style it, whereas in the editor you would have to carefully select all of the text. As for the formatting itself, most of tools you should now find familiar are available to you:

1. At the top of the pane is a condensed version of the Format Bar ([subsection 15.5.2](#)).
2. All of the standard relevant keyboard shortcuts can be used, and indeed most of the commands in the Format menu itself can be used to adjust text formatting.
3. The Ruler is also available for adjusting indents and tab stops.

### Using Styles with Layouts

The Styles button on the far left of the Format Bar will refer to styles provided by the *format itself*, via the Styles format pane ([section 24.5](#)), not any styles in the project. Remember, Compile Formats are self-contained and know nothing about the projects they are applied to.

They also function a little differently to styles in the main editor in that if you modify the formatting of the text at all after applying a style, the assignment to that style will be removed, and it will no longer update when making adjustments to the Style format pane.

### Overriding Text and Notes Formatting

The checkbox by the same name along the bottom of the preview area will toggle whether or not you can format the main text element in the preview area (assuming you are including either **Text** or **Notes** via the checkboxes in the layout list above). There are a few important things to be aware of with this feature:

- Text that is styled in the main text editor will *ignore* any settings made here. Styles in Scrivener are a way of declaring a range of text as special, or anything other than body text if you will.
- It is possible to style text here, and if you require a Word document that has all text styled to “Normal” or “Body”, this would be the best way to do so. You would want to create a style ([subsection 24.5.1](#)) named as needed and then return to the layout that will be generating body text and apply the style to the main text element.



- With the exception of styled text, all text found in the main content area for those items using this layout will be converted to the format you supply here. There are no exceptions. If you need exceptions, refer to the use of styles in the main editor, or the **Format ▶ Preserve Formatting** command, which is designed to exclude text from being altered by this aspect of the compiler.
- If you're looking for a way to adjust indents on body text so they are removed after breaks or headings, refer to the “Settings” tab ([section 24.2.3](#)).

On the right-hand side of the formatting pane you will find a “paintbrush” icon. When formatting override is enabled, this button will bring up a popover with a few exclusions that can be applied to the concept of what is overridden:

**Preserve uncommon alignment** Enabled by default, this option will cause any paragraph alignment other than Left or Justified to be preserved, no matter the alignment of the formatting defined in the sample text above. Left and justified text will always be transformed to match the look of the sample, regardless of this setting.

In most cases you can use Styles to achieve this same effect.

**Preserve tabs and indents** Enable this option to have tab stops and paragraph indents preserved on a per-paragraph basis. This can be useful in cases where you want to generally override the formatting of a section, but the content of that section uses a variety of different indent and tab settings that wouldn't otherwise be applicable to Styles usage. An example of this could be a table of contents section.

### Plain-text

You may be wondering if you can skip this when working with plain-text. Naturally in most cases if you were to set the title to 24pt bold it will not do anything to the compiled document. There are a few notable exceptions that could be of use to you:

- If converting paragraph and indent formatting to whitespace in the Transformations format pane ([section 24.13](#)).
- Where styles are applied to text in the preview area and those styles are set to modify the text somehow, in the Styles format pane ([section 24.5](#)). For example, if you were building an XML format you could create a style that wraps text in a <para> element and then apply that style to body text in the Formatting tab to wrap all body text in these tags.
- Lastly it is also possible for the Markdown-based formats to have rich text converted to Markdown syntax, and thus changes made to the formatting in this pane will have an impact on that conversion where applicable.

### ePub 3 and KF8 (Mobi)

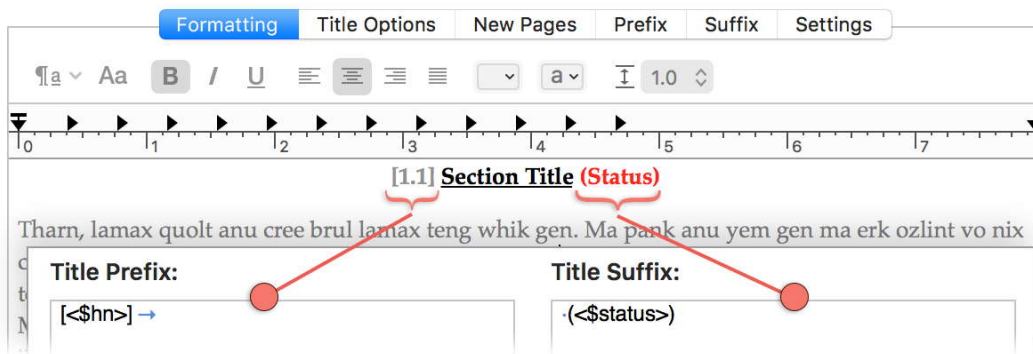
When working with either of these modern and fully semantic file types as your output, the “Formatting” tab will be a little different. Instead of providing full rich text formatting controls, you will only be able to apply styles to text (this will be most useful when assigning styles to HTML Elements ([section 24.9](#))), and overall formatting should be done in the CSS compile format pane ([section 24.7](#)).

### Title Options

The second tab in the layout configuration area provides options for adding a prefix or suffix around the title (or even instead of the title), adjusting the letter case of these elements and other options specific to different file types.

#### The following examples are available in the Extras Pack

If you would like to install the format used to demonstrate the following examples, import the “6-Title Options Examples.scrformat” into your copy of Scrivener (or a test project) from the Extras Pack ([Appendix F](#)).



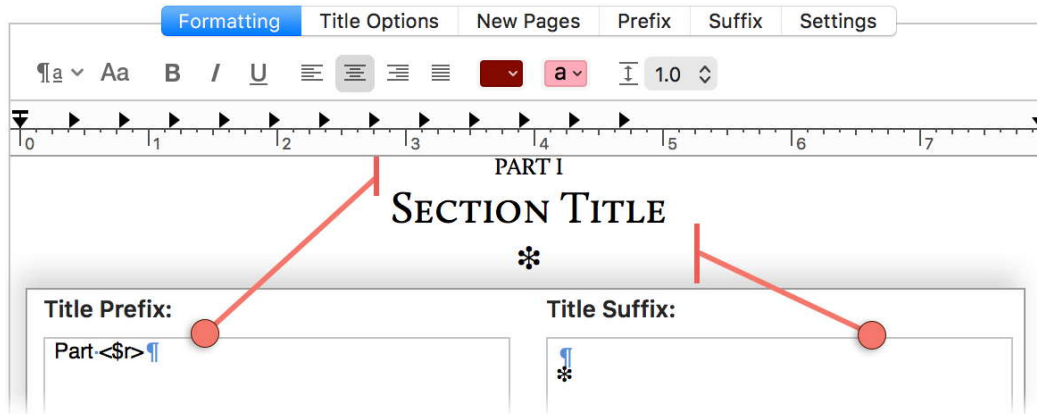
**Figure 24.5:** The title prefix and suffix as applied in the “Formatting” preview tab.

The first example ([Figure 24.5](#)) depicts a single-line title using both a prefix and a suffix. The “Section Title” portion that is bold and underscored is being inserted by the **Title** checkbox for this layout. The **Title Prefix** and **Title Suffix** fields are inserting two different placeholders and some static text:

- The prefix is using the `<$hn>` auto-number placeholder to generate 1.1, 1.1.2, 1.2.4... style numbering to the title. It is surrounded in square brackets and is followed by a tab character. We could use that tab to space the number out from the title further if we wanted.

In the preview tab behind, the prefix has been independently styled to grey text with no underscore.

- The suffix starts with a single space (the small blue dot) followed by the `<$status>` placeholder in parentheses. If the document using the layout has a status of “First Draft”, then we would see it printed as ” (First Draft)”. The suffix has been independently styled to red with no underscore.
- The whole title line—the prefix, title and suffix—has paragraph formatting applied to space it out from the main text below it by 12pts.



**Figure 24.6:** The prefix and suffix can also insert carriage returns.

You can also insert carriage returns into the prefix and suffix fields ([Figure 24.6](#)), and when doing so the formatting controls will be slightly different. Instead of having the entire heading sharing the same paragraph settings as we saw before, each line can have its own paragraph spacing, alignment and other attributes:

- The prefix is typed in normally, but we are using the **Case: Uppercase** setting for the prefix to capitalise it in the preview.
- We could have used the **Title Case: Small Caps** setting to achieve a small caps look on the title, but in all cases it is better to format the text using its native font features for printing small caps, or to use a dedicated small caps font variant, as we have done in the formatting preview area here.
- Finally the suffix inserts another carriage return and a symbol.

One important thing to consider is that the title prefix and suffix entries will be printed *even if the Title is disabled for that row*. It could for example let you use casual titles in the binder, and standard generic numbered titles in the final output. There are a number of layouts in our built-in settings that demonstrate this technique (you could look up the “Part Number Page” layout in the “Modern” format if you want to see a live example), but the setup is simple:

- i. Set the layout so that its **Title** checkbox is disabled.

2. In the “Title Options” tab, add a prefix or suffix.
3. In the “Formatting” preview tab you’ll see whatever you added in the prior step, but not “Section Title” placeholder will be present.

As you have seen from these demonstrations, many of Scrivener’s placeholders (referenced in the [Help ▶ List of all Placeholders...](#) guide) can be used to good effect in these fields. Refer to Using Placeholders in the Prefix and Suffix ([section 24.2.3](#)) for further tips on what can be done.

Beyond the prefix and suffix fields themselves, the following settings are available for modifying the title as a whole, or how those prefix and suffix fields should be handled.

**Insert title as run-in head** When the title (or title suffix) is immediately followed by a standard text block (Main text, Notes, or Synopsis), it will be merged into the first paragraph of that text. In the case of using a suffix on its own line, the suffix will be moved into the first paragraph, not the prefix or title.

When using run-in headings, the font and character attributes of the title will be used to style the title, but its ruler settings will be ignored in favour of the Main Text settings.

This option will not be previewed in the “Formatting” tab, but you will be able to see the results in its respective preview tile, in the compile overview screen.

**Title Case** This setting, along with the **Case** settings that appear below each of the prefix and suffix text boxes, will dynamically adjust the letter case of these title elements. The following options are available to most file types:

- *Normal*: letter case will not be adjusted. However the title elements were typed in will be passed through.
- *Uppercase*: all letters will be converted to UPPERCASE.
- *Small Caps*: this uses faux small caps, by converting all letters to uppercase and then changing the font size on those letters that had been miniscules prior to conversion. The result of this will be visually inferior to a font designed for small caps, so it should only be used if you lack such a font (or one with the typographic features for doing so), or when aiming for file types that cannot use expressive fonts, such as eBook and Web publishing.

This option is naturally only available to formats that use fonts. The setting will render text in uppercase when used with formats like plain-text or the Markdown-based formats.

- *Lowercase*: all letters will be converted to lowercase.

**Title Prefix** Anything typed into this box will be printed directly before either the binder title, or the suffix if the title has been omitted. If you intend to use this to insert the first part of a multi-line title, insert at least one carriage return after the boilerplate text in this box.

**Title Suffix** The contents of this box will be printed directly after the binder title or the prefix. Consequently if you intend for this to display information on a line below the main title, insert at least one carriage return prior to typing anything in.

**Place prefix/suffix inside hashes** Available only to the Markdown-based file types. The title prefix and suffix will ordinarily be placed within the hash marks that Scrivener generates to indicate title depth—thus as part of the heading itself. If you prefer, you can disable the **Place prefix/suffix inside hashes** options to allow text entry outside of the header line itself ([Figure 24.7](#)). Prefix and suffix placement remains literal and directly adjacent to the title element. Thus for proper formatting you will most likely need to insert carriage returns to avoid the prefix/suffix from ending up on the same line as the heading and breaking syntax.



```
### <PREFIX><TITLE><SUFFIX> ###
<PREFIX>### <TITLE> ###<SUFFIX>
```

**Figure 24.7:** The placement of the prefix and suffix around automatically generated hashes with them **inside** and **outside**, respectively.

**Number of hashes** Available only to the Markdown-based file types. Adjusts the number of hashes to use for titles in this section layout. This can be a way of coercing a layout to always print a heading of a certain depth no matter its literal depth in the binder outline. A “Part” section might always want to use a setting of “1”, for example. By default, the “By Level” setting inserts a number of hashes indicating the depth of the item in the outline, regardless of its layout type.

Setting this to “0” will remove the hashes entirely, allowing you to more easily custom format the headings using some other protocol than hashes.

## New Pages

This pane contains settings for adjusting how this Layout will act when a new page (or section break, for those file types that are not built around paper) is generated for it by the Separators compile format pane ([section 24.4](#)).

**Pad top of page with *n* blank lines** Adds the defined quantity of empty lines above the title and section prefix. This will have the effect of pushing the

title down into the page and leave an area at the top blank. Since the setting uses lines this can be used with any file type. This option will be previewed in the section layout's tile, in the compile overview screen.

Even though this option is available to the Markdown-based formats, it should be noted that these systems generally disregard blank lines and they will appear to have no effect in the files they ultimately produce.

**Number of opening words to make uppercase** If your formatting requirements are such that the first few words of the paragraph following a title need to be uppercase, then specify how many words should be set to uppercase with this setting.

**Use small caps** You can also opt to use faked small-caps instead of all upper case, using the checkbox below this setting. This only works with file types that can use font sizes.

**Uppercase even when section is not after a page break** This is the one setting that will trigger even if the section does not generate a page break. Use this if you need to capitalise words following a regular heading or soft break, like a divider or empty line.

**Add “first-letter” span style to the first letter** Available to ePub 3 and KF8 Mobi, this wraps the very first letter of the section in a span with the “first-letter” class assigned to it. This can be used to style drop-caps or other visual effects. You can either create the CSS yourself in the CSS compile format pane ([section 24.7](#)), or you can create a style called “First Letter” and use the WYSIWYG formatting in the Styles pane ([section 24.5](#)). For true drop-caps you will need to use CSS directly as Scrivener itself cannot create floating boxes.

### Setting which page a section falls on

Using the next two options (only available to print and PDF), you can set up common typesetting techniques, such as setting a “part” page to be displayed all by itself on the recto side, with the chapter page following it on the recto side as well and a blank page in between them to do so.

**Always start section on** The new section can be forced to always start on the verso (left) or recto (right) side of the book. This will in some cases cause an empty page to be inserted, in order to keep the chapter on the chosen side.

**Start next section on** If the *following* chunk of text also generates a page break, this setting will control how it behaves if it otherwise doesn't use the prior setting itself.



## Prefix

Use this area to add content to the beginning of the section, directly before the title area (you will need to insert carriage returns into this field if the prefix should be on a separate line). The prefix can be independently formatted using the standard controls provided. With plain text formats, this can be useful for inserting markup around entire sections. The prefix will be previewed in the layout's tile, in the compile overview screen.

You can optionally use the Place prefix after title checkbox to have the prefix inserted directly after the title, starting on its own line. Again however, it will run directly into the following text unless you leave your own carriage returns at the end of the prefix.

Many placeholders can be used in the Prefix and Suffix tabs. Refer to Using Placeholders in the Prefix and Suffix ([section 24.2.3](#)) for further tips on doing so.

### Style prefixes and suffixes in prefixes and suffixes

Style prefix and suffixes will not be applied to text when used in the section prefix or suffix. The style itself will be applied to text in rich text formats that support stylesheets, and as CSS classes in the eBook and HTML formats, but styles as used for plain text and Markdown-based formats will not have much use in these fields.

## Suffix

The suffix tab works similarly to the prefix tab, only inserting the text you provide here at the very end of the section. You will need to add your own separation at the beginning of the suffix field if it should be on its own line. The suffix will be previewed in the layout's tile, in the compile overview screen.

Use the **Place suffix after subdocuments** setting to have the suffix placed after all descendent items in the binder have been printed. In other words, this will place the text at the very end of a container's section of text. A practical example of how this can be used is demonstrated in this user manual, where the links that allow you to return to the nearest chapter break are inserted after each major section. For those creating XML or similar, this is a great way to wrap entire larger sections of text in container elements.

## Using Placeholders in the Prefix and Suffix

Every placeholder (the full list is found in the [Help ▸ List of All Placeholders...](#) menu reference) that you can use in the main text area can also be used in the various prefix and suffix fields found within this pane. Here are a few examples of how that capability can be used:

- While most of the examples in this list are more advanced, it bears repeating that the use of simple auto-number placeholders in the prefix fields are



an ideal way to automatically number sections. Throw “Chapter <\$n>” into your title prefix field, and never worry about keeping the numbers straight in the binder again.

- Those placeholders that pull information from the current document, such as the <\$label> placeholder, will do so for each document that makes use of this Layout, individually.
- You can also *reset* auto-number placeholders. For example you might not want a linear count of figures to be used, but rather for each chapter to have its own figure count starting at one. Since each item in your draft assigned to your chapter Layout will insert a prefix or suffix, that’s a great place to put resets, like: <\$rst\_figure>, which would reset any placeholders using <\$n:figure> in the text.
- With the <\$img...> placeholder you can insert graphics as part of your heading. In a previous example we inserted a special symbol character to print a stylised asterisk, but inserting graphics opens up a great amount of flexibility in how you format your headings. While working within the format designer, images will not be previewed. However image references will show up in the layout tile area of the compile overview screen.

For full documentation on the image placeholder, refer to Image Placeholder Tags ([subsection 15.7.5](#)).

- Compound placeholders can be used. For example you could create a custom “List” type meta-data that provides a few different graphics as named items. One item in that list might be called “Sprouting Seed”. You could then for the chapter folders that should use the “Sprouting Seed” graphic set that custom meta-data option, and then use a placeholder like the following in your prefix or suffix field:

```
<$img:<$custom:ChapterGraphic>>
```

It’s worth noting that images inserted in this fashion will not be displayed as images in the “Formatting” preview area.

- Making use of the <\$include> placeholder, you can include the main text content of a particular binder item into any of the prefix or suffix fields.

When used in this context, you will need to refer to the item by name, like so: <\$include:name of item>. However this raises a problem in that compile Formats are deliberately separate from projects, meaning if you refer to a document by name it is likely the layout will not work in other projects as expected. A way around this is to have “name of item” inserted via *another* placeholder, using the compound form described above.

For example, <\$include:<\$custom:Epigraph>> would insert the text in the “Epigraph” custom metadata field for the item using this placeholder. Presumably, that field would have the binder name of the particular quotation you wished to insert into this location. This keeps the compile format suit-

ably separate from project data.

- Even the title itself can be inserted via the `<$title>` placeholder. This might not seem useful since you can already insert the title with a checkbox, but it might come in handy if for some reason you need to print the title *twice*. The following prefix and suffix would produce a  $\text{\LaTeX}$  code to print the regular title as a chapter heading and then assign a bookmark label for cross-referencing purposes (using a slightly different placeholder that omits spaces):

**Prefix:** `\chapter{`

**Suffix:** `} \label{<$title_no_spaces>}`

## Settings

This final tab will not appear for any plain text or Markdown-based formats.

**Paragraph first line indents** Utilises the common typesetting practice of discarding the first-line indent for any paragraph following a header and/or section break. The calculation for this can be tuned with a set of options below the main checkbox.<sup>1</sup>

- *Do not change:* paragraph formatting will be left alone. This is the default setting for newly created formats.
- *On new pages only:* the first line indent will be remove from the first (unstyled) paragraph found in the binder item using this section layout. Page breaks can be set up in the Separators compile format pane ([section 24.4](#)).
- *At the start of each new document:* this is the default behaviour. Whenever a new document is encountered the first-line indent of the initial paragraph will be stripped out.
- *After empty lines and centered text:* unlike the above two options, this can trigger the removal of the first-line indent from even within an individual section. A full blank line (often used to denote scenes or section breaks), or the presence of any centre-aligned text (such as a scene separator like '#') will trigger it. This can be useful as a compromise between the above two options, where your outline is a casual representation of the reader-accessible structure, and might not strictly conform to section breaks or employ a mix thereof.

---

<sup>1</sup> This adjustment is handled globally, rather than per specific section type in the ePub 3 and KF8 Mobi formats, via the Text Layout compile format pane ([subsection 24.6.4](#)). If you require a more granular approach, consider providing the section layout a CSS class name in this pane and then adjusting the CSS for this section specifically.

**Include in RTF bookmarks** This option is used by all of the word processing formats, which are derived from RTF initially. It is enabled by default for all section layouts, and thus those items not using the “as-is” assignment in your binder will get an RTF bookmark.

These will create handy navigational references throughout the document in word processors, and is also used to cross-reference document links in Scrivener. It may not always be desirable to have bookmarks at every level of your outline, especially if you use Scrivener’s outline feature to break down your book into small blocks. Simply uncheck this to remove the document type and level from the bookmarking feature. If you do this, you will be unable to link to this type of document directly.

**CSS class name** This and the following setting are available to ePub 3 and KF8 Mobi. When a class name (it is up to you to provide a valid class identifier here) is assigned to a section layout, a <div> element will enclose the entire section (including its prefix, title and suffix, if the suffix is not set to fall after subdocuments). This can be used to provide more specific CSS instructions, in the CSS compile format pane ([section 24.7](#)).

**Hide section in e-book** Available to ePub 3 alone. This setting will remove the documents that use this Layout from all forms of navigation in the ebook, including next/prev chapter functions, the internal table of contents, and the automatically generated HTML contents if applicable. It will be referred to as a “non-linear section” in eBook editing programs.

Support for this specification may vary between readers. Some may allow the reader to navigate through these “hidden” sections, but by and large it will not be presented as part of what the reader can navigate to. This can be a useful setting for those writing “choose your own adventure” style stories, or to present extended annotation on a text through the use of links.

[Return to chapter](#) ↗

## 24.3 Script Settings

This pane is only available to the Final Draft (FDX) format. It provides options for configuring a few details that impact how a few elements of a script should be formatted.

**Use default Final Draft screenplay elements** By default, when using the “screenplay” script format, Scrivener’s output will match that of the industry standards used by Final Draft. However if you’ve made changes for your own aesthetic tastes, or are unsure of the formatting in general, you can check this option off to remove Scrivener’s formatting instructions and have Final Draft handle all of the formatting. This could result in a